Lista 2019 – Parte I

2019.1 Seja o sistema linear incerto

$$\delta(x) = A(\alpha)x,$$

sendo $A(\alpha)$ uma matriz politópica, isto é,

$$A(\alpha) = \sum_{i=1}^{N} \alpha_i A_i, \quad \alpha \in \Lambda_N$$

cujos vértices A_i, $i = 1, \ldots, N$ são conhecidos e Λ_N é o simplex unitário dado por

$$\Lambda_N \triangleq \left\{ \xi \in \mathbb{R}^N : \sum_{i=1}^{N} \xi_i = 1,\ \xi_i \geq 0,\ i = 1, \ldots, N \right\}$$

O operador $\delta(\cdot)$ representa a derivada ($\delta x(t) = d(x(t))/dt$) no caso de sistemas a tempo contínuo e o operador deslocamento unitário ($px(k) = x(k+1)$) no caso de sistemas a tempo discreto. Elabore uma rotina em matlab que desenhe o lugar das raízes da matriz $A(\alpha)$ testando diversos valores de α (malha fina). A rotina deve receber como entrada as matrizes A_i em formato célula ($A\{i\}$) e um *flag* indicando se é um sistema contínuo ou discreto. A malha fina deverá contemplar, no mínimo, os seguintes conjuntos de pontos

- 1000 pontos uniformemente distribuídos dentro do politopo. Para gerar esses pontos, use o código apresentado no apêndice.
- 100 pontos (igualmente espaçados) para cada segmento de reta entre quaisquer par de vértices (i, j) do politopo.
- 150 pontos uniformemente distribuídos dentro do subpolitopo formado por cada tripla (i, j, k) de vértices do politopo. Observação: não se aplica a sistemas com dois e três vértices.

Plote também os eixos real e imaginário e no caso de sistemas discretos, o círculo unitário. Como saída, a rotina deve fornecer o máximo valor da parte real (do módulo) dos autovalores, bem como o ponto α^* associado. Entregue o código fonte e dois exemplos gerados aleatoriamente de modo que o autovalor de maior parte real (maior módulo) não aconteça nos vértices ($\alpha_i = 1$, $\alpha_j = 0,\ i \neq j$).

2019.2 Faça um programa para gerar sistemas lineares discretos com matrizes aleatórias incertas $A(\alpha)$, $B(\alpha)$, $C(\alpha)$ e $D(\alpha)$ politópicas com $A(\alpha)$ Hurwitz estável e que a maior parte real dos autovalores seja aproximadamente igual a d. Dados n (número de estados), m (número de entradas), p (número de saídas), N (número de vértices) e $d < 0$, o programa deve retornar as matrizes A, B, C, D contendo os vértices em formato celular, além de também fornecer o valor das normas $\mathcal{H}_2$ e $\mathcal{H}_\infty$ do sistema (a norma $\mathcal{H}_2$ pode ser calculada considerando $D(\alpha) = 0$). Entregue o código fonte e um exemplo gerado aleatoriamente de modo que o autovalor de maior parte real seja aproximadamente -0.05 (o lugar das raízes desse exemplo deverá ser fornecido).

2019.3 Seja a LMI dependente de parâmetros

$$A(\alpha(t))P(\alpha(t)) + P(\alpha(t))A(\alpha(t))' - \dot{P}(\alpha(t)) + B(\alpha(t))B(\alpha(t))' < 0$$

(a) Usando o Lema de Finsler, obtenha uma condição equivalente com variáveis de folga.

(b) Usando novamente o Lema de Finsler, obtenha uma LMI equivalente à condição obtida em (a) com mais variáveis de folga.

2019.4 Determine as normas $\mathcal{H}_2$ e $\mathcal{H}_\infty$ do sistema

$$\begin{cases} \dot{x} = -x + 3w \\ y = x \end{cases}$$

usando a definição. Compare os valores determinados com os obtidos por meio do comando `norm` do matlab.

2019.5 Seja o sistema

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bw(t) \\ y(t) &= Cx(t) + Dw(t) \end{aligned} \tag{1}$$

com as seguintes matrizes

$$A = \begin{bmatrix} 0 & 1 \\ -16 & -1.2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 2 \end{bmatrix}, \quad D = 0$$

(a) Considerando $w(t) = \sin(\omega t)\exp(-0.1t)$, $t \in [0, 10]$, faça um programa que computa a relação entre a energia do sinal de saída $y(t)$ e a energia do sinal de entrada $w(t)$. O cômputo da energia de um sinal pode ser feito utilizando a rotina *trapz*, por exemplo

```
w=3;
T=0.01;
t=0:T:10;
sinal=sin(w*t).*exp(-0.1*t);
energia=sqrt(trapz(sinal.^2)*T);
```

(b) Compute a norma $\mathcal{H}_\infty$ do sistema usando LMIs (apresente o código) e depois teste o programa elaborado para os seguintes valores de $\omega \in \{3.8, 3.85, 3.9, 3.95, 4.0, 4.05, 4.10, 4.15, 4.20\}$.

(c) Desenhe o Diagrama de Bode e determine a norma $\mathcal{H}_\infty$ por inspeção, isto é, determinando o valor do pico (use a ferramenta de *zoom* para aumentar a precisão da inspeção). O valor obtido é próximo do computado no item (b)? Se não, o que poderia ser feito para diminuir a diferença?

(d) Substitua o sinal de entrada pelo impulso unitário e compute a norma $\mathcal{H}_2$ por meio da determinação da energia da saída.

(e) Compute a norma $\mathcal{H}_2$ do sistema usando LMIs (apresente o código) e compare o resultado com o obtido no item (d).

Referências

[1] A. Moeini, B. Abbasi, and H. Mahlooji. Conditional distribution inverse method in generating uniform random vectors over a simplex. *Communications in Statistics – Simulation and Computation*, 40(5):685–693, 2011.

Apêndice

Código que pode ser utilizado para encontrar pontos randômicos uniformemente distribuídos dentro do simplex unitário. Fonte: [1]

```
N=3 % número de vértices
x(1)=1-rand^(1/(N-1))
for k=2:N-1
    x(k)=(1-sum(x(1:k-1)))*(1-rand^(1/(N-k)));
end
x(N)=1-sum(x(1:N-1))
```